**Experiment 9**

**Simulate the following File allocation strategies -**

**(a) Sequenced (b) Indexed (c) Linked**

**(a) Sequenced**

**Program:**

#include<stdio.h>

void main()

{

int n,i,j,b[20],sb[20],t[20],x,c[20][20];

printf("Enter no.of files:");

scanf("%d",&n);

for(i=0;i<n;i++)

{

printf("Enter no. of blocks occupied by file%d",i+1);

scanf("%d",&b[i]);

printf("Enter the starting block of file%d",i+1);

scanf("%d",&sb[i]);

t[i]=sb[i];

for(j=0;j<b[i];j++)

c[i][j]=sb[i]++;

}

printf("Filename\tStart block\tlength\n");

for(i=0;i<n;i++)

printf("%d\t %d \t%d\n",i+1,t[i],b[i]);

printf("Enter file name:");

scanf("%d",&x);

printf("File name is:%d",x);

printf("length is:%d",b[x-1]);

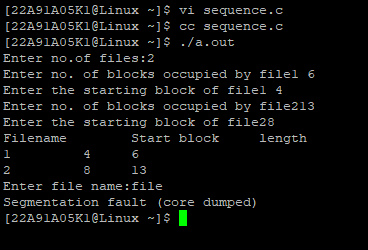
printf("blocks occupied:");

for(i=0;i<b[x-1];i++)

printf("%4d",c[x-1][i]);

}

**Output:**



**(b) Indexed**

**Program:**

#include<stdio.h>

void main()

{

int n,m[20],i,j,sb[20],s[20],b[20][20],x;

printf("Enter no. of files:");

scanf("%d",&n);

for(i=0;i<n;i++)

{

printf("Enter starting block and size of file%d:",i+1);

scanf("%d%d",&sb[i],&s[i]);

printf("Enter blocks occupied by file%d:",i+1);

scanf("%d",&m[i]);

printf("enter blocks of file%d:",i+1);

for(j=0;j<m[i];j++)

scanf("%d",&b[i][j]);

}

printf("\nFile\t index\tlength\n");

for(i=0;i<n;i++)

{

printf("%d\t%d\t%d\n",i+1,sb[i],m[i]);

}

printf("\nEnter file name:");

scanf("%d",&x);

printf("file name is:%d\n",x);

i=x-1;

printf("Index is:%d",sb[i]);

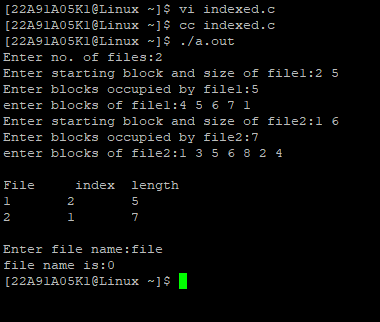
printf("Block occupied are:");

for(j=0;j<m[i];j++)

printf("%3d",b[i][j]);

}

**Output:**

****

**(b) Linked**

**Program:**

#include<stdio.h>

struct file

{

char fname[10];

int start,size,block[10];

}

f[10];

void main()

{

int i,j,n;

printf("Enter no. of files:");

scanf("%d",&n);

for(i=0;i<n;i++)

{

printf("Enter file name:");

scanf("%s",&f[i].fname);

printf("Enter starting block:");

scanf("%d",&f[i].start);

f[i].block[0]=f[i].start;

printf("Enter no.of blocks:");

scanf("%d",&f[i].size);

printf("Enter block numbers:");

for(j=1;j<=f[i].size;j++)

{

scanf("%d",&f[i].block[j]);

}

}

printf("File\tstart\tsize\tblock\n");

for(i=0;i<n;i++)

{

printf("%s\t%d\t%d\t",f[i].fname,f[i].start,f[i].size);

for(j=1;j<=f[i].size-1;j++)

printf("%d--->",f[i].block[j]);

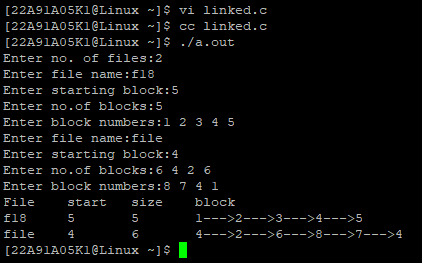
printf("%d",f[i].block[j]);

printf("\n");

}

}

**Output:**

****

**Experiment 10**

**Write a C program that illustrates two processes communicating using shared**

**memory**

**Program:**

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/shm.h>

#include<string.h>

int main()

{

int i;

void \*shared\_memory;

char buff[100];

int shmid;

shmid=shmget((key\_t)2345, 1024, 0666|IPC\_CREAT);

printf("Key of shared memory is %d\n",shmid);

shared\_memory=shmat(shmid,NULL,0);

printf("Process attached at %p\n",shared\_memory);

printf("Enter some data to write to shared memory\n");

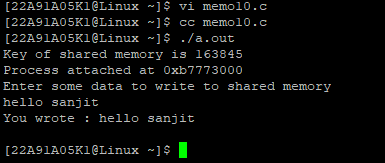
read(0,buff,100);

strcpy(shared\_memory,buff);

printf("You wrote : %s\n",(char \*)shared\_memory);

}

**Output:**

****